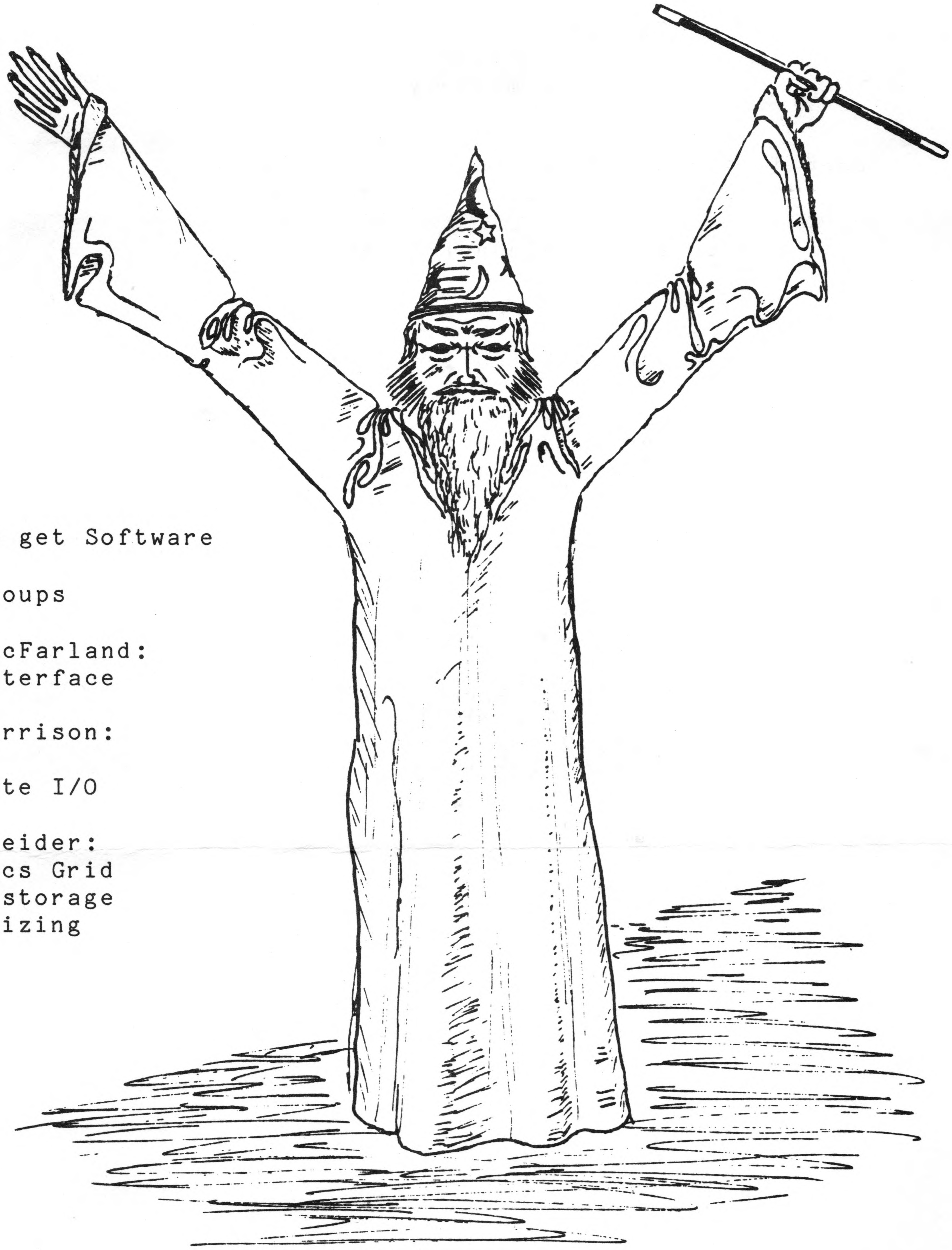
THE Exidy Monitor

January 30th, 1980 Volume I, Issue VIII $1.00



Where to get Software

Other Groups

Craig MacFarland:
  TTY interface

Roger Harrison:
  SOS
  Cassette I/O

Art Schneider:
  Graphics Grid
  BASIC storage
  Randomizing

Michael Vachon

THE EXIDY MONITOR

printed monthly

John T. Vachon, President and Chief Editor
Marilyn Barry, Editorial Assistant

Cover art by Michael J. Vachon

Major contributions by:

Roger S. Harrison Jr.
Craig MacFarland
Art Schneider

"The Exidy Monitor"
is sponsored by

COMPUTER MART OF MASSACHUSETTS
1395 MAIN STREET
WALTHAM, MA 02154
(617) 899-4540

COMPUTER MART, INC.

The Exidy Monitor
Volume I, Issue 8

Hello. Sorry about this issue being so late. But here is some information that you may find helpful:

Quality Software of Reseda, CA, is selling a book which gives a lot of information on how the Sorcerer software is put together and can be taken apart. This includes information on the routines in the Sorcerer's monitor.

MacLean Design Graphics in Toronto has a machine-language program that renumbers BASIC programs and another program for merging BASIC programs.

Some corrections from last issue: In "Ants in Sugar II" line 30 should read: `30 J=INT(RND(1)*128+128)` and line 100 should read: `100 P=INT(RND(1)*1024)+1`

David Riggs has been nominated for the "Helping Hint of the Year" award for his discovery of how to change the line length in BASIC, in the last issue of this newsletter.

Charles Grebert is looking for some business software. In particular, he is looking for the Radio Shack Budget Manager already transferred to the Sorcerer. If you have any information please call him at 648-3984.

Following you will find a list of Sorcerer User groups that are known to us.

Sorcerer User's Group
P.O. Box 144
Docaster, Vic. 3108
Australia
Frank Schuffelen, Editor
$ 7.00 annual subscription
$10.00 overseas

The Sorcerer's Apprentice
1530 Washtenaw
Ann Arbor, MI 48104
Dave Bristor, Publisher
Send SASE for copy

Sorcerer User's Newsletter
792 Laurie Avenue
Santa Clara, CA 95050
Steven J. Long, Editor
$10.00 annual subscription
$17.00 overseas

Sorcerer's Source
P.O. Box 43
Audubon, PA 19407
Terry Laudereau, Editor

The Southern California Sorcerers Users Group
5218 1/2 Pine Ave.
Maywood, CA 90207
Ron Miranda, Publisher
$.75 per issue

1395 Main St., Waltham, MA 02154 (617) 899-4540

HILLTOP ELECTRONIC ASSEMBLERS

8 ADAMS STREET
BILLERICA, MASS. 01821
TELEPHONE - (617) 667-5560

JONH T. VACHON, EDITOR
THE EXIDY MONITOR
1395 MAIN STREET
WALTHAM, MASS. 02154

DEAR JOHN:

SINCE I PURCHASED MY SORCERER LAST DECEMBER, I HAVE OFTEN WONDERED HOW USERS, SUCH AS MYSELF, COULD TIE IN A TTY AND USE IT AS A PRINTER. WE ALL KNOW THAT THE SORCERER IS NOT PREPARED TO DO THIS TASK, AND FOR GOOD REASON. WHO COULD POSSIBLY WANT A NOISY, SLOW AND DIRTY MACHANICAL MONSTER BEING TIED TO THEIR ELABORATE COMPUTER? THEN ANSWER IS SIMPLE. ** ANYONE WHO OWNS ONE AND CAN'T AFFORD THE HIGH PRICED SPREAD **

DURING THE PAST FEW MONTHS, I HAVE TRIED MANY WAYS TO SOLVE THE TTY PROBLEM, AND FINALLY HAVE COME UP WITH A WAY WHICH WILL NOT ONLY LET YOU USE THE TTY FOR A PRINTER BUT ALSO ALLOW YOU TO USE THE TTY KEYBOARD AS AN INPUT TO THE SORCERER. MOST OF THE CREDIT BELONGS TO KEN BARBIER WHO WROTE AN ARTICLE ON THE SUBJECT IN KILOBAUD. I MODIFIED SOME OF HIS SCHEMATICS, WIRE WRAPPED THE CIRCUIT AND PACKAGED IT IN A 6 BY 3 BOX.

USING THE SORCERERS MONITOR COMMAND SE S=20, WE CAN FOOL THE TTY INTO THINKING IT SEES 110 BAUD. THE COMMAND SE O=P, SETS THE OUTPUT PORT TO PARALLEL WHICH IN TURN DRIVES THE PRINTER. THE COMMAND SE I=P SETS THE INPUT PORT TO PARALLEL WHICH WILL ALLOW THE USE OF THE TTY'S KEYBOARD.

IF ANYONE IS INTERESTED IN GETTING ONE OF THESE DEVICES BUILT AND TESTED FOR THEIR SORCERER, PLEASE CONTACT ME AT THE ABOVE ADDRESS. A SET OF PRINTS ARE ALSO AVAILABLE FOR ANYONE WHO WANTS TO BUILD IT HIMSELF.

THIS LETTER WAS PRINTED ON A TTY USING THE SORCERER AS ITS COMPUTER.

SINCERELY,

C. CRAIG MACFARLAND
PRESIDENT

SOS

by Roger S. Harrison

Here is a little utility program for tape recovery or inspection.

SOS is not a polished program, but it does the job for desperate tape recovery purposes.

SOS!

```
EXIDY Z-80 ASSEMBLER
   ADDR  OBJECT    ST #

                   0001 ;    S O S   Sorcerer Output Scanner
                   0002 ;          OPTION D  DUMPS EVERY TAPE CHAR TO SCREEN,
                   0003 ;                    INCL NULLS, CRC, ETC.
                   0004 ;          OPTION P  PRINTS TEXT CHARS
                   0005 ;          BOTH OPTIONS DISPLAY O/P BYTE COUNT IN HEX
                   0006 ;          AT TIME JOB IS HALTED OR ETX (03H) FOUND.
                   0007 ;          by ROGER S. HARRISON, JR.
  '>0000           0008 SOS        EQU     $         ;DUMP OR PRINT TAPES
  '0000  CDB1E9    0009            CALL    0E9B1H    ;INIT SCREEN
  '0003  216100'   0010            LD      HL,MSG    ;Print or Dump
  '0006  CDBAE1    0011            CALL    0E1BAH    ;MSGOUT
  '0009  CD18E0    0012 KYBD       CALL    0E018H    ;KEYBRD
  '000C  28FB      0013            JR      Z,KYBD-$ ;WAIT FOR I/P
  '000E  CDF0E9    0014            CALL    0E9F0H    ;ECHO
  '0011  213400'   0015            LD      HL,SW+01H ;JR DISP BYTE
  '0014  3613      0016            LD      (HL),PRINT-SW-2H ;DISPL
  '0016  FE44      0017            CP      'D'       ;DUMP RAW DATA
  '0018  2002      0018            JR      NZ,GO-$ ;NO, PRETTY PRINT
```

CONT

```
'001A  3600     0019        LD    (HL),0H ;USE CURR PC VALUE AT SW.
'001C  0601     0020 GO     LD    B,01H   ;DRV 1
'001E  F6FF     0021        OR    0FFH    ;INIT CASSETTE
'0020  D3FD     0022        OUT   (0FDH),A
'0022  CD24E0   0023        CALL  0E024H  ;CMOTON
'0025  110000   0024        LD    DE,00H  ;INIT COUNT
'0028  CD05E2   0025        CALL  0E205H  ;CRLF
'002B  2100F1   0026 TOP    LD    HL,0F080H+128 ;RETAIN TOP ROW
'002E  CD0FE0   0027 NEXT   CALL  0E00FH  ;INTAPE
'0031  281F     0028        JR    Z,END-$ ;FINI
'0033  1813     0029 SW     JR    PRINT-$ ;JR OR NO JR
'0035  77       0030        LD    (HL),A  ;BYTE TO SCREEN
'0036  FE03     0031        CP    03H     ;ETX ?
'0038  2818     0032        JR    Z,END-$ ;FINI
'003A  23       0033        INC   HL
'003B  13       0034        INC   DE      ;COUNT BYTES
'003C  3EF8     0035        LD    A,0F8H  ;OFF SCREEN ADDR
'003E  BC       0036        CP    H       ;SCREEN FULL?
'003F  20ED     0037        JR    NZ,NEXT-$ ;NO
'0041  3E0C     0038        LD    A,0CH   ;FF
'0043  CD1BE0   0039        CALL  0E01BH  ;DO IT!
'0046  18E3     0040        JR    TOP-$   ;REFILL SCREEN
'0048  FE03     0041 PRINT  CP    03H     ;ETX ?
'004A  2806     0042        JR    Z,END-$ ;FINI
'004C  CD1BE0   0043        CALL  0E01BH  ;VIDEO
'004F  13       0044        INC   DE      ;COUNT BYTES
'0050  18DC     0045        JR    NEXT-$
'0052  218000'  0046 END    LD    HL,SPC
'0055  CDBAE1   0047        CALL  0E1BAH  ;SPC OVER
'0058  CDE8E1   0048        CALL  0E1E8H  ;BYTE COUNT OUT
'005B  0601     0049        LD    B,01H   ;DRV 1
'005D  CD27E0   0050        CALL  0E027H  ;CMOTOF
'0060  C9       0051        RET
'0061  0D       0052 MSG    DEFB  0DH     ;CR
'0062  2020     0053        DEFW  '  '
'0064  2020     0054        DEFW  '  '
'0066  2020     0055        DEFW  '  '
'0068  2053     0056        DEFW  'S '
'006A  204F     0057        DEFW  'O '
'006C  2053     0058        DEFW  'S '
'006E  207E     0059        DEFW  '~ '
'0070  2020     0060        DEFW  '  '
'0072  4475     0061        DEFW  'uD'
'0074  6D70     0062        DEFW  'pm'
'0076  20       0063        DEFB  ' '
'0077  6F72     0064        DEFW  'ro'
'0079  20       0065        DEFB  ' '
'007A  5072     0066        DEFW  'rP'
'007C  696E     0067        DEFW  'ni'
'007E  7420     0068        DEFW  ' t'
'0080  2020     0069 SPC    DEFW  '  '
'0082  2020     0070        DEFW  '  '
'0084  00       0071        DEFB  0H
ERRORS=0000

END         0052 GO         001C KYBD       0009
MSG         0061 NEXT       002E PRINT      0048
SOS         0000 SPC        0080 SW         0033
TOP         002B
```

1200 BAUD CASSETTE IO

By: Roger S. Harrison, Jr. 12/79

The purpose of this program is to describe a method of performing 1200 BAUD cassette input/output on the Exidy Sorcerer computer. The general method used is to POKE a SAVE or LOAD monitor command into the Monitor Work Area, MWA, so that it appears to the Sorcerer as though the command had been entered from the keyboard. In each case, the appropriate monitor command (SAVE or LOAD) is then executed by using the USR(0) function.

All monitor commands can be executed this way if you know the monitor entry points (these have been widely published). However, some commands, like SET, are easier to implement by direct POKE statements; another like TEST, utterly destroys all of RAM (perhaps as a program security measure if a password is not entered correctly after a given number of tries).

The "difficulty" of using the SAVE command in this way is due to the fact that SAVE requires absolute hexadecimal addresses. That in itself would not create much difficulty if only BASIC would leave its arrays in one place - it doesn't (fortunately). Since simple variables, including some strings, fall between the BASIC statement region and the array region, any declared arrays move up six bytes for each new simple variable referenced. So the address of the array area is simply not absolute...UNLESS certain simple variables are delcared prior to accessing the very first array's absolute address. More precisely, if you want your array to "stay put", no new simple variables should be declared between PEEKing locations 1B9H and 1BAH (absolute address of first array at that time) and executing the USR function call to SAVE or LOAD. In the accompanying program, this includes all variables used in subroutines at 760 and 800 as well as "U" in the USR call. If you do use any previously undeclared simple variables during that critical period, the absolute start address of the array (S in accompanying program) will be too small by six bytes for each new variable. Note that the first declared array must be your IO buffer for all input and output, and it must be numeric.

For that price, here is what you get:

1. 1200 BAUD cassette input/output
2. All the monitor's file capabilities, i.e., named, CRC checked data blocks , dual cassette support, etc.
3. Ability to skip unwanted files by using filenames when executing SAVE and LOAD.
4. Because the array name is not saved on cassette, the IO buffer name when reloading does not have to be the same as the array from which it was saved. It must be identically dimensioned however.
5. Goodbye to variable data in DATA statements! If you can dimension it, you can read/write it.

Since insufficient information is sometimes more frustrating than no information, here is some background on how to use the "debugging" aids built into the program as PRINT statements. The only key parts of the program required for use in your application program are the statements 782-788 for LOAD, statements 790-798 for SAVE, and the "predefined" variables as described above.

Hex locations 1B9H and 1BAH (LLHH as usual) contain the address of the two byte name of the first dimensioned array. This is BASIC's pointer to the array and must be unchanged between placing the absolute address in variable S, and executing the SAVE or LOAD command via THE USR(0) call. S+2 points just beyond the array name to its two byte length, so the name is not written to tape. This 2 byte field is accessed and put into L. The array end address is therefore S+L+1. The start address S and end address are converted to hex strings by the routine at 800 for insertion into the LOAD or SAVE command. The entire command CMD$ is POKEd into the MWA buffer by the routine at 760.

For debugging, CMD$ appears three ways: in ASCII values and in character as it would have been typed, and in hex from location 0000H to 000FH. Both S and L are printed in ASCII.

Since I sometimes, not always use 48K, routine at 740 determines the machine size and initializes the MWA address, AD, accordingly. Also I use drive 1 for my program so DV$ is set to "2" (must be string).

Just using the LOAD facility of this technique is very helpful. Since my Startrek space ship characters (SHIPS) always resides at FD00H, I let the BASIC Startrek program load the graphic characters by setting:

```
    CMD$ = "LO SHIPS" + CHR$(13),
```

then executing a routine like 760 which POKEs this CMD$ into the MWA buffer. If SHIPS is conveniently the very next block of data on tape, the filename SHIPS could be omitted.

I have been using a less restrictive Z-80 assembler language adaption of this technique for some time now. Duplication and enhancement of this technique is both permitted and encouraged.

```
100 REM EXTRACT STARTING ADDRS AND LGH OF FIRST DIMed ARRAY
110 REM FOR CREATING A MONITOR 'SAVE' & 'LOAD' COMMAND
120 REM     by Roger S. Harrison, Jr.   (201) 385-8904
160 CLEAR 100
170 AZKI$="0123456789ABCDEF"
178 REM MUST DECLARE AT LEAST ALL VARIABLES BETWEEN PEEK(441&42)
179 REM AND THE USR(0) FUNCTION CALL !
180 S=0:L=0:V=0:U=0:Y1=0:Z1=0
181 N1$="":Z1$="":CD$="":CMD$="":NM$="":DV$="":S$="":E$=""
190 M=1:N=1
199 REM IO BUFFER MUST BE FIRST DIMENSIONED,NUMERIC ARRAY
200 DIM IO(M,N)
210 PRINT:CD$="":INPUT"LO or SA or END";CD$
211 IF CD$="END"THEN END
215 IF CD$="LO" OR CD$="SA" THEN 220
216 GOTO210
220 FORI=0TOM:FORJ=0TON:IO(I,J)=(I+1)*(J+2):NEXTJ:NEXTI
735 NM$="A":DV$="2"
740 AD=16273:IF PEEK(-4095)=127 THEN AD=32657
745 IF PEEK(-4095)=191 THEN AD=48927-65536
750 FORI=0TO59:POKE(AD+I),32:NEXTI
758 GOTO780
760 FORI=0TOLEN(CMD$)-1:U=ASC(MID$(CMD$,I+1,1)):POKE(AD+I),U
770 NEXTI
772 FORI=0TO19:POKE(0+I),PEEK(AD+I):PRINTPEEK(AD+I);:NEXTI:PRINT
774 PRINTCMD$
775 RETURN
780 IF CD$="SA" THEN790
782 GOSUB890
784 V=PEEK(441)+256*PEEK(442)+2:GOSUB800:S$=N1$
786 CMD$="LO "+NM$+" "+DV$+" "+S$+CHR$(13)
788 GOSUB760:POKE260,138:POKE261,231:U=USR(0):GOTO850
790 S=PEEK(441)+256*PEEK(442)+2:V=S:GOSUB800:S$=N1$
792 L=PEEK(S)+256*PEEK(S+1)+1
793 PRINT"S="S,L,PEEK(S),PEEK(S+1)
794 V=S+L:GOSUB800:E$=N1$
795 PRINT"S,L="S,L,PEEK(441),PEEK(442)
796 CMD$="SA "+NM$+" "+S$+" "+E$+" "+DV$+CHR$(13)
798 GOSUB760:POKE260,56:POKE261,230:U=USR(0):GOTO850
800 N1$=""
805 Y1=INT(V/16):Z1=V-Y1*16
810 FORI=0TOZ1:Z1$=MID$(AZKI$,I+1,1):NEXTI
820 N1$=Z1$+N1$
840 IFY1=0THENRETURN
845 V=Y1:GOTO805
850 FORI=0TOM:FORJ=0TON:PRINTIO(I,J);:NEXTJ:NEXTI:PRINT
860 GOTO210
890 FORI=0TOM:FORJ=0TON:IO(M,N)=0:NEXTJ:NEXTI:RETURN
900 END
READY
```

'LOAD'
MONITOR ENTRY

'SAVE'
MONITOR ENTRY

8 Melanie Lane
Mattapoisett Ma 02739
617 758 4229
Jan. 2, 1980

John T Vachon
Computer Mart of Mass
1395 Main St
Waltham Ma 02154

Dear John

Enclosed are several items for Sorcerer users that you may include in the EXIDY MONITOR. The topics are diverse and will be discussed as follows:

1. Graphic Grid Paper.
2. Basic sub. to Randomize a set of numbers.
3. Basic program storage.

I've referenced several previous EXIDY MONITOR issues.

1. The GRAPHIC GRID

The Graphic Grid is a special graph paper I prepared for the Sorcerer. Copy attached. The grid proportions match both the individual characters and full screen layout. You will find it convienient for several graphic layouts and designs; these include:

A. CHARACTER GRAPHIC DESIGN
Use one 8*8 dark line large block for each graphic character. With a pencil, darken the small boxes to create character design. The correct proportions make the GRID very handy for multi-character designs. Then use CHAR program to determine data values (User Tape Vol X). The starting address for your POKE statements to load in the graphic data values can be found by:

```
ADDRESSS = -2048 + 8 * ASCII#.
```

B. FULL SCREEN LAYOUT
Use the small boxes for each character on the vidio screen. The Graphic GRID is numbered with ROW (0-29) and COLumn (0-63) marks to agree with Bill Hann's function (see Vol. 1 Issue V page 3) or as modified:

```
DEF FNC(COL) = -3968 + 64*ROW + COL
DEF FNR(ROW) = -3968 + 64*ROW + COL
```

C. GAME BOARD SQUARE DESIGN
Use the inner darkened large character size 5 COL, 3 ROW Square Box (emphase on square) to design special game board squares. You can easily generate game boards with multiples of these 15 Char. (5 x 3) squares. Typical 8 x 8 game boards are easy; 10 x 10 boards require more care.

The game board square leads to comments on the GRAPHIC GRID design. Each small box on the grid represents the size ratio of a character on the screen: 5:3. Specifically I used 5mm x 3mm in constructing the grid. The upper 64 x 30 zone on the grid approximates the actual total screen size on a Video-100. !!! SUGGEST !!! You COPY the grid before using.

2. BASIC SUB TO RANDOMIZE A SET OF INTERGERS

Randomizing a set of intergers has many uses; such as placing a specific number of things or events in a random order. I'm using this technique in a sport match schedule program. Another example is the shuffling a deck of 52 cards. The set of intergers from 1 to 52 must be placed in a random order. The card shuffle differs from a dice throw in that each number (1-6) has the same chance each time. In cards the 'number' is used with each deal. The attached Sub. is well documented (illustrating good documentation if you have the memory). Included is program that calls the Sub. along with a demonstration run.

3. BASIC PROGRAM STORAGE

Till now little has been said in the Exidy Monitor about how the Sorcerer stores a Basic program. Knowledge about this could be usefull for such things as self modifing code techniues like a line renumbering program. The following is not a complete discussion of the topic but rather a view of "tip of the iceburg".

With a little PEEKing or monitor DUmping you can confirm that your Basic program starts at memory location 01D5 Hex or 469 decimal (after a LIST or RUN). The storage is line oriented; each line has 5 components in storage detailed below:

1. Starting address of next line in memory; 2 bytes.
   byte 1 + byte 2 * 256 = address (Lo-Hi Z80 format)
2. Line number used Basic program; 2 bytes
   byte 1 + byte 2 * 256 = Line number
3. Key word code 1 byte
   ASCII value above 128 (see Vol 1, Issue 1). Codes are the same as the one stroke basic key graphics.
4. Remainder of Basic program line
   Non key words use ASCII value numbers at one byte per character (<=127). Line numbers in GOTO etc are in ASCII. Key words and operators use one byte codes >128. Exception - graphics in quotes.
5. 0 zero end of line mark. CR & LF not stored
   tripple zero 000 3 bytes end the program.

This is illustrated with a short (but useless) example:

PROGRAM

```
100 REM AAA
300 DATA 2,3.5
555 END
```

SORCERER STORAGE

•••••••• •••••••

LINE	HEX		DECIMAL		COMMENT
	LOC.	VALUE	LOC	VAL	
100	01D5	DF	469	223	(Address of next line
	D6	01	470	1	223 + 1 * 256 = 479)
	D7	64	471	100	(Line number in program
	D8	00	472	0	100 + 0 * 256 = 100)
	D9	8F	473	143	REM Key word code (graphic W)
	DA	20	474	32	space ASCII code
	DB	41	475	65	A " "
	DC	41	476	65	A " "
	DD	41	477	65	A " "
	DE	00	478	0	end of line mark - a zero
300	01DF	EB	479	235	(Address of next line
	01E0	01	480	1	235 + 1 * 256 = 491
	E1	2C	481	44	(Line number in program
	E2	01	482	1	44 + 1 * 256 = 300
	E3	83	483	131	DATA Key word code (graph 4)
	E4	20	484	32	space ASCII code
	E5	32	485	50	2 " "
	E6	2C	486	44	, " "
	E7	33	487	51	3 " "
	E8	2E	488	46	. " "
	E9	35	489	53	5 " "
	EA	00	490	0	end of line mark
555	01EB	F1	491	241	Address of next line
	EC	01	492	1	
	ED	2B	493	43	(line number
	EE	02	494	2	43 + 2 * 256 = 555
	EF	80	495	128	END Key word code (graphic 1)
	01F0	00	496	0	end of line mark
	F1	00	497	0	(end of program -
	F2	00	498	0	two zeros)

The topic of Basic storage is covered in more detail in Personal Computing, March 1979 p24. The article details Basic storage in the PET computer that also uses Microsoft Basic. The Sorcerer shares many similar storage formates. Starting location and a few Key word ASCII codes differ (THEN = 162 for the Sorcerer; 167 for PET). The article includes a line renumbering program; this would be most welcome for the Sorcerer; especially in Z80 Assembly.

These notes on Basic program storage leave more questions than answers. I hope this stimulates someone to answer the following:

1. The memory from 256 to 468 (0100H-01D4H) apperars to contain Basic control parameters. Usefull changes may be possible. Some are loaded by the Basic Pack directly; others are loaded by a Tape load or program entry. So far we Know (from Monitor articles):

HEX	DEC.	Comment
0103	259	Z80 Jump for USR(x) - C3
0104	260	Low address for USR(x)
0105	261	High address
0142	322	Print line length (64 default) (see Vol 1 Issue VII, David Risg)

What about the other locations???

2. Where is the Basic Stack located?
3. What memory is reserved for 'free string space'?
4. Where are values for program varriables stored?
5. How about the input line buffer?
6. Finally how do we determine avaliable memory not used by Basic other than 0-256 that could be used for extended machine language routines and related storage.

In closing I'll note This Letter was prepared using the Sorcerer WORD PROCESSOR PAC and the IDS 440 Paper Tiger printer. Both are a delight to use.

Yours truly,

Art Schneider

Art Schneider

Enclosed
1. The GRAPHIC GRID
2. BASIC SUB - Randomise integers (with Demo. run).

```
list 10000

10000 REM ***
10020 REM          SUB...TO RANDOMIZE A SET OF INTERGERS
10040 REM ***
10060 REM Variables passed to sub
10080 REM     N8   # of intergers to rnd
10100 REM     s8   starting value
10120 REM     b8   Step value in interger set
10140 REM Variables returned
10160 REM     P(  ) intergers in a rnd order
10180 REM Assumed  P(  ) was in DIM & RND set with RND(-x)
10200 REM     -x a negitive number
10220 REM Variables used
10240 REM     Z8,Y8,X8
10260 REM***
10300 FOR Z8 = 1 TO N8     :REM Load in the intergers
10320 :  P(Z8) = Z8*B8 + S8-B8
10340 NEXT Z8
10360 FOR Z8 = 1 TO N8     :REM RND the sequence
10380 :  Y8 = P(Z8)
10400 :  X8 = INT(RND(1) * (N8-Z8+1) + Z8)
10420 :  P(Z8) = P(X8) : P(X8)=Y8
10440 NEXT Z8
10500 RETURN
10600 REM
10700 REM ******************
READY
```

```
list

100 REM RANDI  by Art Schneider  Jan. 2,1980
120 REM        Demo. to Randomize a set values
130 REM
140 CLEAR 100 : CLEAR
160 PRINT "Randomize your selected set of numbers"
180 PRINT: INPUT "Seed the RND sequence: Type any number";SEED
200 SEED = -ABS( INT(SEED))                :REM must be neqitive
210 IF INT(SEED/2 ) = SEED/2 THEN SEED = SEED -3 :REM & odd.
215 PRINT "Random Seed =";SEED
220 REM  see Vol I Issue V pg 12 EXIDY MONITOR
240 SEQ = RND( SEED)                       :REM to select sequence
260 PRINT
280 REM  Following input Var. S8,N8,B8 used by subrutine 10000
300 INPUT "Starting number in your set......";S8
320 INPUT "Number of values in your set.....";N8
340 INPUT "Step value in your set...........";B8
360 DIM P( N8 )                            :REM P stores RND values
380 GOSUB 10000                            :REM to RND value set
400 LAST = S8 + B8*(N8-1)                  :REM Last value in set
420 PRINT : PRINT "OK  now your";N8;"numbers in RANDOMIZED ";
440 PRINT "SEQUENCE;"
460 PRINT "between";S8;"and";LAST;"( incremented by";B8;" ):"
480 PRINT
500 FOR I = 1 TO N8 : PRINT P( I ); : NEXT I
520 PRINT : PRINT : PRINT : END
600 REM
620 REM ************************************************
640 REM
660 REM
BREAK
READY



run
Randomize your selected set of numbers

Seed the RND sequence: Type any number? 2840
Random Seed =-2843

Starting number in your set......? 100
Number of values in your set.....? 30
Step value in your set...........? 5

OK  now your 30 numbers in RANDOMIZED SEQUENCE;
between 100 and 245 (incremented by 5 ):

 150  190  185  105  205  110  160  155  170  175  220  140
 245  180  135  215  165  125  210  230  235  145  195  240
 200  115  225  120  130  100


READY
```

SORCERER SYSTEMS AND PERIPHERALS
AVAILABLE FROM COMPUTER MART, INC.
1395 MAIN STREET, WALTHAM, MA 02154
(617) 899-4540

PSC 16K Sorcerer	1295.00
PSC 32K Sorcerer	1395.00
PSC 48K Sorcerer	1495.00
(All Sorcerers include Operation and BASIC manuals plus CMI diagnostic software and literature)	
Leedex Video 100 12" Monitor	149.00
Leedex Video 100 12" Monitor with Sorcerer purchase	104.00
Leedex Video Monitor cable	10.00
Integral Data Systems IP-125 Printer w/ Printer control	838.00
Cable for IP-125 and IP-225 printers	50.00
Integral Data Systems 440 Printer w/ Forms control	995.00
Graphics and 2K buffer options for the 440	199.00
AJ Selectric Terminal Printer	1295.00
Cable for IDS-440 printer or AJ printer	45.00
S-100 Expansion Cabinet with power supply and mother board	350.00
Fan option for Expansion Box	25.00
16K Memory Expansion Kit	99.00
Installed and Guaranteed	135.00
32K Memory Expansion Kit	195.00
Installed and Guaranteed	245.00
Sorcerer Operations Manual	9.95
Sorcerer Standard BASIC ROM PAC or Development PAC (manuals inc.)	99.00
Sorcerer Word Processor ROM PAC (manual included)	199.00
Sorcerer BASIC manual, Technical manual, Word Processing manual, or Development Tour manual; each:	11.95
Sorcerer EPROM PAC for 2708, 2716, or 2732 tri-volt EPROMS	49.00
Serial Data Cassette Cable	65.00
D-30 Data Cassettes	1.50
Data Cassette Recorder	50.00
Tek-Com Accoustic Coupler Modem TC-3006 (1 yr warranty)	199.95
RMP Industries Music Box for Sorcerer (includes software)	49.95
Vista Dual Drive, DD Mini-Disk System w/ Disk BASIC E and CP/M	1299.00
Extra Drive, Cabinet, and Power Supply	475.00
Micropolis DD Mod I disk System w/ Disk BASIC and MDOS	795.00
Additional drive w/ cabinet and power supply	545.00
CP/M for Micropolis w/ Sorcerer Drivers	145.00
Micromation Doubler Interface for 8" drives	495.00
CP/M and BASIC E for Doubler w/ Sorcerer drivers	125.00
8" Shugart Drive w/ Cabinet and Power Supply	795.00
Single Drive Cable for 8" disk	35.00
Double Drive Cable for 8" disk	45.00
Micromation Dual Disk, DD MEGABOX w/ CP/M and BASIC E	2495.00
Micromation Single Disk, DD system w/ CP/M and BASIC E	1450.00
Model 6 Newtech Music Board, assembled and tested	59.95
Full line of S-100 Boards (Price list available soon)	

Prices effective January 15, 1980

Commercially Produced Sorcerer Software on Cassette

CLINIC:	biorhythm, diet, longevity	(24K)	14.95
BEAT THE HOUSE:	blackjack, roulette, craps, slot machine	(32K)	14.95
TOUCHDOWN:	a fast-paced football simulation	(16K)	14.95
STATPLOT:	statistical plotting with linear regression	(16K)	14.95
SUB:	an arcade game with sound effects from a radio	(8K)	9.95
X AND O:	different levels of TIC-TAC-TOE challenges	(8K)	9.95
DOODL:	convert your Sorcerer to a super Etch-A-Sketch	(8K)	9.95
KIDDO:	calculator programs using giant characters	(8K)	9.95
BIO:	the best biorhythm program available for home	(8K)	9.95
FAMOUS AMERICANS:	combines anagrams with action graphics	(16K)	9.95
BLACKJACK:	play against the computer or up to six players	(16K)	9.95
U.S. MAP:	state capital quizes utilizing US and state maps	(16K)	9.95
ATOM:	deductive reasoning game for locating atoms	(16K)	9.95
PERSONALITY TEST:	analysizes personality, plots results	(16K)	9.95
COMPUCARDS:	gives up to 200,000 fortunes using playing cards	(16K)	9.95
FUNCTION PLOT:	plot functions using cartesian or polar coord.	(16K)	9.95
AWARI:	ancient game of logic from the NIM family	(8K)	9.95
CHASE:	escape from robots using mines and barriers	(8K)	9.95
GRAPHIC GAMES II:	6 Creative Computing games	(8K)	7.95
Z80 DISASSEMBLER:	get assembly code from machine language	(8K)	9.95
KNOW YOUR SORCERER I:	tutorial instructions on 5 keybd. levels	(8K)	7.95
KNOW YOUR SORCERER II:	further tutorials incl. graphics	(8K)	7.95
KNOW YOUR SORCERER III:	advanced programming tutorials & Tech.	(16K)	14.95

Sorcerer User's Group Volumes each: 6.50

Vol I:	(8K)	AMAZE, LUNAR, ROTE
Vol II:	(8K)	LUCAS, RSP, CHECK
Vol III:	(8K)	DEPTH, REVRS, HIQUE
Vol IV:	(8K)	BIO, MIDAG, MANDL
Vol V:	(16K)	HANGM, HPAWN
Vol VI:	(32K)	HANG2, HANG3
Vol VII:	(32K)	WARPX, TREK1
Vol VIII:	(32K)	WARPZ, TREK1
Vol IX:	(8K)	BANER, BOXOP
Vol X:	(32K)	DEMO, CHAR
Vol XI:	(8K)	DUMBT, DUMB2
*Vol XII:	(32K)	LIFE, OTHLO, BRIDG
Vol XIII:	(8K)	PLOT, HISTO
Vol XIV:	(8K)	LIFE6, CHARX
Vol XV:	(8K)	SHOOT, FPLOT, FDRAW

Note: Volume XII (stared) not available yet

Sorcerer Software on Diskette (Specify type of Disk System)

Datafile: Data base manag. using Micropolis BASIC (Mod I/II)		45.00
Microsoft extended BASIC ver. 5 (requires CP/M)		300.00
Microsoft Macro-80 (requires CP/M)		149.00
Microsoft BASIC Compiler incl. MACRO-80 (requires CP/M)		350.00
Microsoft Fortran (requires CP/M)		500.00
Microsoft COBOL (requires CP/M)		750.00
Xitan Disk BASIC (requires CP/M)		159.00
Xitan Disk Z-TEL Text Editor, Text Output Processor, Macro Assembler or Linker (requires CP/M);	each:	69.00
Xitan Disk Z-BUG Debugger (requires CP/M)		89.00
CP/M User's Library Volumes I through XXXV	each disk:	10.00
C BASIC II (requires CP/M)		95.00
Structured Systems Q SORT (requires C BASIC)		95.00
Selector III-C2 Data Base Management (requires C BASIC)		345.00
Creative Comp. "ADVENTURE" (requires MicroSoft BASIC, 48K)		25.00

Prices effective January 15, 1980

COMPUTER MART, INC.

"The Exidy Monitor"
back-issues

☐ Issue 1: $ .25: printer control, useful routines
☐ Issue 2: 1.00: PEEK, POKE, USR, memory test
☐ Issue 3: 1.00: D-PAC, Z-80 I, Misc info, questionnaires
☐ Issue 4: 1.00: Z-80 II, Software library, changes
☐ Issue 5: 1.00: Wide Printer driver, RND, INVRS, graphics, TESTS
☐ Issue 6: 1.00: IBM Selectric driver, "Plane"
☐ Issue 7: 1.00: Letters, Bit Manipulation, Keyboard status

Total for back issues: $________
shipping, $.25 per issue or $1.00 max: $________

Total purchase: $________

Merchandise Order

QTY	DESCRIPION	COST EACH	COST TOTAL

SUB TOTAL PURCHASE:	
MA residents add 5% sales tax:	
Shipping:	1.50
TOTAL:	
Total for back issues from above:	
GRANDE TOTAL:	

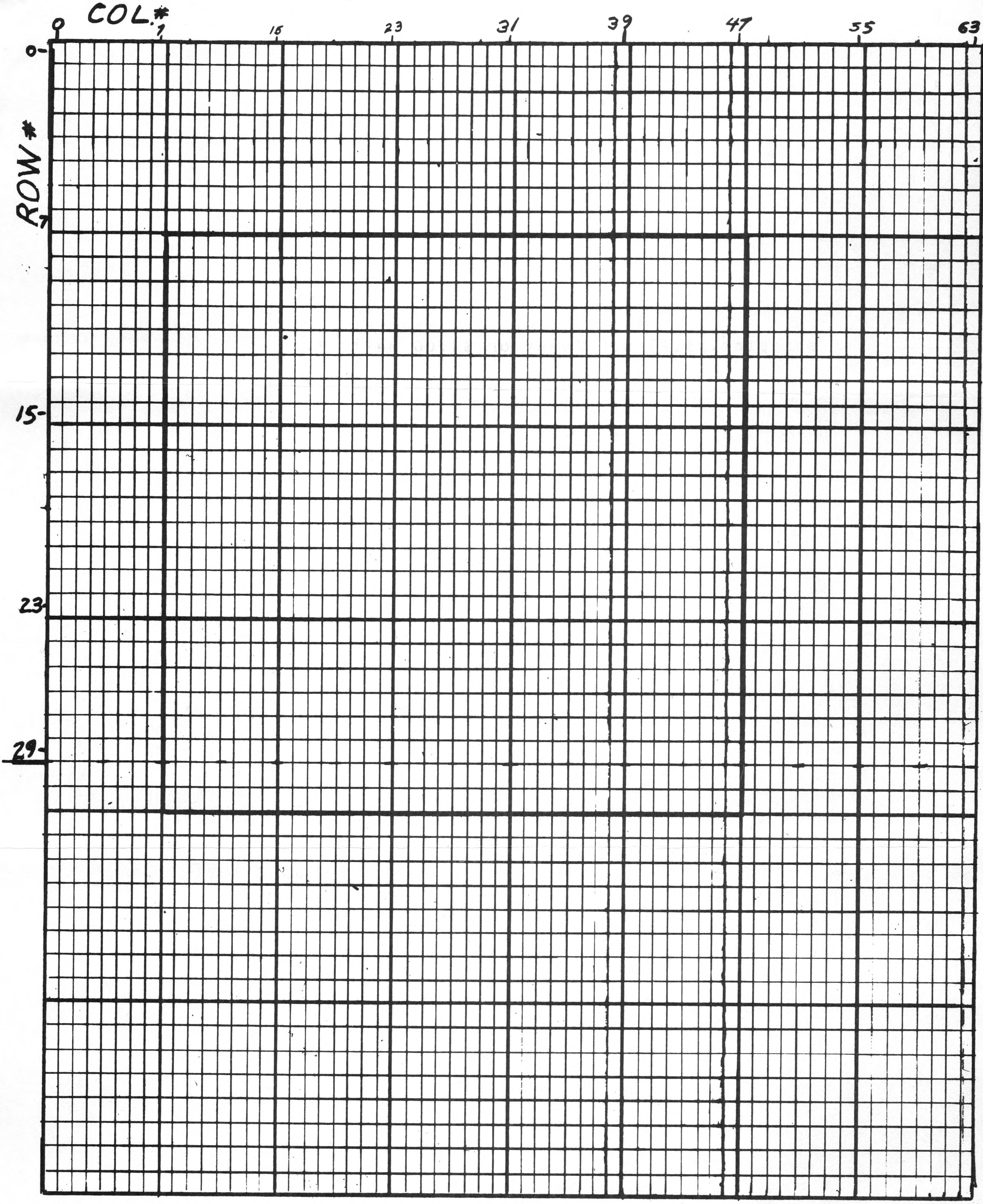
Enclosed is $______ for the above checked off items.
☐ CASH, CHECK, or MONEY ORDER payable to Computer Mart, Inc.
($25.00 minimum on charges:)
☐ VISA/BANK AMERICARD ☐ MASTERCHARGE
Card #________________________ Exp. date:________________

NAME:
ADDRESS:
CITY: STATE: ZIP:

Please send to address indicated.

1395 Main St., Waltham, MA 02154 (617) 899-4540

CHARACTER GRAPHIC (8 × 8) : VIDIO [(0-63) X (0-29)]



GRAPHIC GRID by Art Schneider 30 NOV 79

for SORCERER VIDIO SIZE RATIO : ROW/COL = 5/3